UJM at INEX 2009 XML Mining Track

Christine Largeron, Christophe Moulin, Mathias Géry

UJM at INEX 2009 XML Mining Track*

Christine Largeron and Christophe Moulin and Mathias Géry

Université de Lyon, F-42023, Saint-Étienne, France
CNRS UMR 5516, Laboratoire Hubert Curien
Université de Saint-Étienne Jean Monnet, F-42023, France
{christine.largeron,christophe.moulin,mathias.gery}@univ-st-etienne.fr

Abstract. This paper reports our experiments carried out for the INEX XML Mining track 2009, consisting in developing categorization methods for multi-labeled XML documents. We represent XML documents as vectors of indexed terms. The purpose of our experiments is twofold: firstly we aim to compare strategies that reduce the index size using an improved feature selection criteria CCD. Secondly, we compare a thresholding strategy ($MCut$) we proposed with common $RCut$, $PCut$ strategies. The index size was reduced in such a way that the results were less good than expected. However, we obtained good improvements with the $MCut$ thresholding strategy.

1 Introduction

This paper describes the participation of Jean Monnet University at the INEX 2009 XML Mining Track. For the categorization task (or classification), given a set of categories, a training set of preclassified documents is provided. Using this training set, the task consists in learning the classes descriptions in order to be able to classify a new document in the categories.

One main difference in the collection of documents provided in INEX 2009 relatively to INEX 2008 lies in the overlapping of the categories and in their dependencies [1]. When each document belongs to one and only one category in INEX 2008, it can belong to several categories in INEX 2009. With the imbalance between the categories, their overlapping poses new challenges and gives opportunities for design machine learning algorithms more suited for XML documents mining.

In this article, we focus on the selection of the set of classes that will label a document for this multi-label text categorization. We explore two approaches. The first one uses a binary classifier which considers one category against the others. The algorithm returns two answers (yes or no) used to decide whether the document belongs or not to this category. In that case, the selection of a set of words characteristic of the category can be essential for improving the performance of the algorithm. Our first contribution to Inex 2009 consists in an

* This work has been partly funded by the Web Intelligence project (région Rhône-Alpes: http://www.web-intelligence-rhone-alpes.org)

improvement of the selection criteria that we have introduced in INEX 2008 and which permitted to get the best results of the competition while reducing the index size [2]. The second approach uses a multi-label classifier which considers simultaneously all the categories. Given a document, the classifier returns a score (*i.e.* a numerical value), for each category. In the context of single label classification in which one and only one class must be attributed to each document the decision rule is obvious: it consists to return the class corresponding to the best score. On the contrary, in multi-label categorization, this approach raises the question of the number of classes that must be assigned to each document. In this article, we propose a thresholding strategy for selection of candidate classes and we compare it with the commonly used methods $PCut$ and $RCut$ [7, 4].
In the aim of introducing our notations, a brief presentation of the vector space model (VSM [5]), used to represent the documents, is given in section 2, the selection features criteria are defined in the following section. The thresholding strategy for selection of candidate classes is presented in section 4, while the runs and the obtained results are detailed in sections 5 and 6.

2 Document model for categorization

Vector space model, introduced by Salton et al. [5], has been widely used for representing text documents as vectors which contain terms weights. Given a collection D of documents, an index $T = \{t_1, t_2, ..., t_{|T|}\}$, where $|T|$ denotes the cardinal of T, gives the list of terms (or features) encountered in the documents of D. A document d_i of D is represented by a vector $\overrightarrow{d_i} = (w_{i,1}, w_{i,2}, ..., w_{i,|T|})$ where $w_{i,j}$ is the weight of the term t_j in the document d_i defined according the TF.IDF formula :

$$w_{i,j} = tf_{i,j} \times idf_j$$

with $tf_{i,j} = \frac{n_{i,j}}{\sum_l n_{i,l}}$ where $n_{i,j}$ is the number of occurrences of t_j in document d_i normalized by the number of occurrences of terms in document d_i and $idf_j = \log \frac{|D|}{|\{d_i : t_j \in d_i\}|}$ where $|D|$ is the total number of documents in the corpus and $|\{d_i : t_j \in d_i\}|$ is the number of documents in which the term t_j occurs at least one time.

3 Criteria for features selection

3.1 Category Coverage criteria (CC)

In the context of text categorization, the number of terms belonging to the index can be exceedingly large and all these terms are not necessarily discriminant features of the categories. It is the reason why, it can be useful to select a subset of T giving a more representative description of the documents belonging to each category. For this purpose, we proposed in a previous work a selection features criteria, called coverage criteria CC and based on the frequency of the documents containing the term [2]. Let $f_j^k = \frac{df_j^k}{|c_k|}$ be the frequency of documents belonging

to c_k and including t_j where $df_j^k = |\{d_i \in c_k : t_j \in d_i\}|, k \in \{1, ...r\}$ is the number of documents in the category c_k in which the term t_j appears and $|c_k|$ is the number of documents belonging to c_k. The higher the number of documents of category c_k containing t_j, the higher f_j^k, CC is defined by:

$$CC_j^k = \frac{df_j^k}{|c_k|} * \frac{f_j^k}{\sum_k f_j^k} = \frac{(f_j^k)^2}{\sum_k f_j^k}$$

If the value of CC_j^k is high, then t_j is a characteristic feature of the category c_k.

3.2 Difference Category Coverage criteria (CCD)

The previous criteria considers only the coverage of the category by one term but it does not take into account the coverage of the other categories. The difference of category coverage permits to overcome this drawback. Thus, the Category Coverage Difference CCD is defined by :

$$CCD_j^k = (CC_j^k - CC_j^{\bar{k}})$$

$$CC_j^{\bar{k}} = \frac{(f_j^{\bar{k}})^2}{f_j^k + f_j^{\bar{k}}}$$

with $f_j^{\bar{k}} = \frac{df_j^{\bar{k}}}{|D|-|c_k|}$ and $df_j^{\bar{k}} = |\{d_i \in D \wedge d_i \notin c_k : t_j \in d_i\}|, k \in \{1, .., r\}$.
As previously, if the value of CCD_j^k is high, then t_j is a characteristic feature of the category c_k.

The CC and CCD criteria can be used for multi-label text categorization by binary classifier. For each category and consequently each classifier, they permit to reduce the index to the set of words which are the most characteristic of this category.

4 Thresholding strategies

When a multi-label algorithm is used in multi-label text categorization, one score $\phi(\overrightarrow{d_i}, c_k)$ is produced by the classifier for each document-category pair (d_i, c_k). Given these scores, the problem consists to determine the set of classes $L(d_i)$ which must be attributed to each document d_i.
To solve this problem, different approaches have been proposed, which consist in applying a threshold to the scores returned by the classifier. In $RCut$ method, given a document d_i, the scores $(\phi(\overrightarrow{d_i}, c_k), k = 1, .., r)$ are ranked and the t top ranked classes are assigned to d_i. The value of the parameter t can be either specified by the user or learned using a training set. In $PCut$ method, given a category c_k, the scores $(\phi(\overrightarrow{d_i}, c_k), i = 1, .., |D|)$ are ranked and the n_k top ranked documents are assigned to the class with:

$$n_k = P(c_k) * x * r$$

where $P(c_k)$ is the prior probability for a document to belong to c_k, r, is the number of categories, and x is a parameter which must be estimated using a training set. A review of these methods can be found in [7, 4]
In $PCut$ as well as in $RCut$, the performance of the classifier depends on the value of the parameter (t or x). The main advantage of the thresholding method proposed in this article is that the threshold is automatically fixed.

This method, called $MCut$ (for Maximum Cut) is based on the following principle, explained graphically. Given a document d_i, the scores $(\phi(\overrightarrow{d_i}, c_k), k = 1, .., r)$ are ranked in decreasing order. The sorted list obtained is noted $S = (s(l), l = 1, .., r)$ where $s(l) = \phi(\overrightarrow{d_i}, c_k)$ if $\phi(\overrightarrow{d_i}, c_k)$ is the lth highest value in S.

Then, a graph of the scores in their decreasing order is drawned (*i.e.* $s(l), l = 1, .., r$ in function of l). The value t retained as threshold is the middle of the maximum gap for S: $t|(s(t)+s(t+1))/2 = Max\{(s(l)+s(l+1))/2, l = 1, .., r-1\}$

The clusters assigned to d_i are those corresponding to a score $\phi(\overrightarrow{d_i}, c_k)$ higher than $t : L(d_i) = \{c_k \in C/\phi(\overrightarrow{d_i}, c_k) > t\}$.

For instance, the graphs of the scores $(s(l), l = 1, .., r)$ versus l obtained for two documents: d_1 (on the left side) and d_2 (on the right one) are presented in figure 1. Using $MCut$, the document d_1 is assigned to one class (on the left) while the document d_2 is assigned to three classes (on the right).

$MCut$ is also compared with the $RCut$ strategy in figure 1. In $RCut_1$ (resp. $RCut_2$), the t parameter is set up to 1 (resp. 2). For the document d_1, the same set of classes is affected by $RCut_1$ and $MCut$, while $RCut_2$ assigns one class more. In the second case, the set of affected classes is different. While d_2 belongs to three classes with $MCut$ strategy, it is associated to one class (resp. two classes) with the $RCut_1$ (resp. $RCut_2$) strategy.

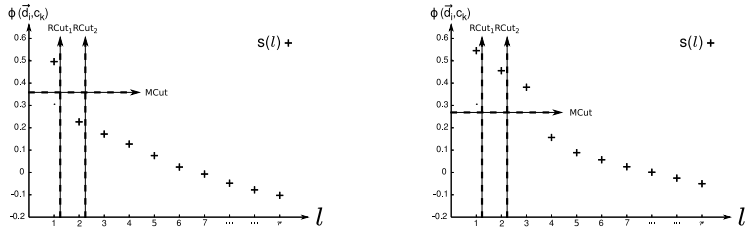

Fig. 1. Illustration comparing RCut and MCut thresholdling strategies.

5 Experiments

5.1 Collection INEX XML Mining

The XML Mining collection is composed of about 54 889 XML documents of the Wikipedia XML Corpus. This subset of Wikipedia represents 39 categories, each corresponding to one subject or topic. This year, the collection is multi-label and each document belongs to at least one category. In the XML Mining

Track, the training set is composed of 20% of the collection which corresponds to 11 028 documents. On the training set, the mean of the number of category by document is 1.46 and 9 809 documents belong to only one category.

5.2 Pre-processing and categorization

The first step of the categorization approach that we propose, consists in a pre-processing of the collection. It begins by the construction of the list all the terms (or features) encountered in the documents of the collection. This index of 1 136 737 terms is built with the LEMUR software[1]. The Porter Algorithm [3] has also been applied in order to reduce different forms of a word to a common form. After this pre-processing, it still remains a large number of irrelevant terms that could degrade the categorization, e.g.: numbers (7277, -1224, 0d254c, etc.), terms with less than three characters, terms that appear less than three times, or terms that appear in almost all the documents of the training set corpus. After their deletion, the index size is reduced to 295 721 terms on all the documents and it will be noted T. Depending on the category c_k, T_k will correspond to the index only composed of terms that appear in documents of the category c_k. In order to reduce the index size, we also define $T_{k_{1000}}$ as the index of the category c_k composed of the most characteristic terms of category c_k according to the CCD criteria introduced in section 3. If CCD^k_{1000} corresponds to the best thousandth score obtained with the CCD criteria for the category c_k, $T_{k_{1000}}$ is composed of all terms t_j for which CCD^k_j is higher than CCD^k_{1000}. All the indexes definitions are summarized in the table 1.

Index		Definition
T	=	$\{t_j \in d_i \mid d_i \in D\}$
T_k	=	$\{t_j \in d_i \mid d_i \in D \wedge d_i \in c_k\}$
$T_{k_{1000}}$	=	$\{t_j \in T_k \wedge CCD^k_j \geqslant CCD^k_{1000}\}$

Table 1. Summary of all defined indexes.

The second step is the categorization step itself. The Support Vector Machines (SVM) classifiers are used for the categorization. SVM was introduced by Vapnik for solving two classes pattern recognition problems using Structural Risk Minimization principal[6]. In our experiments, the SVM algorithm available in the Liblinear library[2] has been used.

In the XML Mining Track, the final score ($score(d_i, c_k)$) assigned to a document d_i for the category c_k has to be included in $[0, 1]$ and has to be higher than 0.5 if this document belongs to the category c_k. When the SVM is used as a multi-label classifier (noted $multi-label$), it provides a score $\phi(\overrightarrow{d_i}, c_k)$ for each pair document - category (d_i,c_k). In that case, the final score $score(d_i, c_k)$

[1] Lemur is available at the URL http://www.lemurproject.org

[2] http://www.csie.ntu.edu.tw/∼cjlin/liblinear/ - L2 loss support vector machine primal

associated to d_i and c_k corresponds to $\phi(\overrightarrow{d_i}, c_k)$ normalized. So, the final result, given d_i, is a set of classes ordered by relevancy. When the SVM is used as a binary classifier (noted $uni - label$), it provides, given a category c_k, two scores ($\phi_k(\overrightarrow{d_i}, c_k)$ and $\phi_k(\overrightarrow{d_i}, \bar{c_k})$). In that case, the score $score(d_i, c_k)$ equals 1 if $\phi_k(\overrightarrow{d_i}, c_k) > \phi_k(\overrightarrow{d_i}, \bar{c_k})$ and equals 0 otherwise. The final result is a set of unordered classes.

5.3 Submitted runs

Run	SVM	Index	Thresholding Strategy	Set of classes
lahc_1_baseline	$multi - label$	T	-	$singleton$
lahc_2_binary	$uni - label$	T	-	$unordered$
lahc_3_binary_lk	$uni - label$	T_k	-	$unordered$
lahc_4_binary_lk_1000	$uni - label$	$T_{k_{1000}}$	-	$unordered$
lahc_5_max	$multi - label$	T	$MCut$	$ordered$
lahc_6_pcut	$multi - label$	T	$PCut$	$ordered$
lahc_7_rcut_1	$multi - label$	T	$RCut_1$	$ordered$
lahc_8_rcut_2	$multi - label$	T	$RCut_2$	$ordered$

Table 2. Summary of our XML Mining experiments

In the context of multi-label text categorization, our aim was to evaluate on one hand the influence of the features selection on the performance of the binary classifier (runs *lahc_2*, *lahc_3*, *lahc_4*) and on the other hand the impact of the thresholding strategies on the multi-label classifier (runs *lahc_5*, *lahc_6*, *lahc_7*, *lahc_8*). We have submitted 8 runs based on different indexes and thresholding strategies, summarized in table 2. Given the SVM score $\phi(\overrightarrow{d_i}, c_k)$, the first 4 runs uses the *unordered* method to compute the final score $score(d_i, c_k)$ and the last 4 runs the *ordered* one.

The first run (*lahc_1*) corresponds to the baseline. This run only assigns one category for each document. This category corresponds to the highest score provided by the multi-label SVM classifier ($multi - label$).

In order to evaluate the influence of the selection features on the performances, the three next runs consider the SVM as a binary classifier ($uni - label$) employing different indexes. The index T (resp. T_k, $T_{k_{1000}}$) is tested with the second run (*lahc_2*) (resp. *lahc_3*, *lahc_4*). The binary classifier can assign no category to a document. In that case, for *lahc_3* and *lahc_4* runs, the category c_k provided by the baseline run (*lahc_1*) is affected to the document.

The last four runs exploit the different thresholding strategies detailed in section 4. The fifth run (*lahc_5*) uses the $MCut$ strategy. The run *lahc_6* exploits the $PCut$ strategy with x equals to the number of documents in the test set divided by the number of categories.The run *lahc_7* (resp. *lahc_8*) applies the $RCut$ strategy using a parameter t fixed to 1 (resp. 2).

6 Experimental results

All the results are summarized in table 3. The common criteria (ACC, ROC, PRF and MAP) used for the XML Mining evaluation are presented. In order to rank the runs, we introduce the Mean criteria that corresponds to the average obtained

for the micro and the macro value for ACC, ROC and PRF. We will firstly discuss results of our baseline. Secondly we will detail the results concerning the selection features criteria and finally those which exploit a thresholding strategy.

Participant	Run	Macro ACC	Micro ACC	Macro ROC	Micro ROC	Macro PRF	Micro PRF	MAP	Mean
lhc	lahc_5_max	0,968	0,952	0,936	0,934	0,549	0,578	0,788	0,820
lhc	lahc_7_rcut_1	0,974	0,962	0,938	0,935	0,531	0,564	0,788	0,817
lhc	lahc_6_pcut	0,973	0,961	0,927	0,925	0,548	0,563	0,748	0,816
lhc	lahc_8_rcut_2	0,959	0,933	0,903	0,906	0,515	0,528	0,788	0,791
xerox	nxQ.3.merge.tfidf	0,975	0,964	0,753	0,767	0,579	0,605	0,678	0,774
xerox	netxQ.4.plus.tfidf	0,974	0,963	0,748	0,765	0,571	0,600	0,679	0,770
xerox	nxQ.4.merge	0,974	0,963	0,748	0,765	0,571	0,600	0,679	0,770
peking	3	0,963	0,948	0,842	0,850	0,480	0,519	0,702	0,767
peking	2	0,963	0,948	0,842	0,850	0,480	0,518	0,702	0,767
peking	1	0,962	0,947	0,842	0,850	0,478	0,516	0,702	0,766
granada	nb_with_links_sub	0,952	0,934	0,802	0,820	0,500	0,530	0,642	0,756
granada	nb_sub	0,951	0,933	0,803	0,820	0,496	0,527	0,641	0,755
lhc	lahc_1_baseline	0,974	0,962	0,721	0,743	0,531	0,564	0,685	0,749
granada	orgate_with_links_sub	0,848	0,819	0,928	0,927	0,316	0,360	0,725	0,700
lhc	lahc_3_binary_lk	0,967	0,950	0,619	0,629	0,334	0,355	0,407	0,642
granada	orgate_sub	0,754	0,678	0,925	0,922	0,253	0,263	0,730	0,632
lhc	lahc_2_binary	0,971	0,958	0,600	0,613	0,289	0,323	0,393	0,626
lhc	lahc_4_binary_lk_1000	0,965	0,947	0,585	0,596	0,252	0,279	0,330	0,604
wollongon	bpts2.f1.r3	0,913	0,892	0,625	0,619	0,192	0,218	0,138	0,576
wollongon	bptsext.f1a.r3	0,131	0,160	0,558	0,561	0,072	0,103	0,100	0,264
wollongon	bptsext.f1.r3	0,038	0,055	0,632	0,623	0,071	0,102	0,208	0,253
wollongon	bpts2.f1a.r3	0,038	0,055	0,598	0,599	0,071	0,102	0,125	0,244
wollongon	bptsext.map.r3	0,137	0,141	0,506	0,513	0,065	0,096	0,192	0,243
wollongon	bpts2.map.r3	0,115	0,123	0,511	0,510	0,070	0,101	0,129	0,238

Table 3. Summary of all XML Mining results sorted by Mean.

Baseline results (run: 1). On table 3, our baseline results are quite good if we compare them to other participant results. As this run limits the number of affectations, it also reduces the number of errors and that is why this is our best run for the ACC criteria. As we only consider a binary score (*unordered*), ROC and MAP criteria are not very good since they take into account the order of returned categories. The PRF criteria, that combines precision and recall, is not very high. That means that we should have a correct precision, but a very low recall because this run considers only one category by document.

Features selection runs (run: 2, 3, 4). Runs 2, 3 and 4 aim to evaluate the influence of the index size using different binary classifiers for each category. As we can see on table 3, all these runs are worse than our baseline for all evaluation criteria. On average on the different evaluation criteria, run 3 is better than runs 2 and 4.

We can conclude that the index reduction is not satisfying and has to be improved. The first idea is to come back to the strategy proposed in INEX 2008 and which consists to define a global index by union of the categories' indexes $\cup_{k \in C} T_{k_{1000}}$. The second idea is to use a different number of terms depending on the category.

Thresholding strategy runs (run: 5, 6, 7, 8). All the runs, that use thresholding strategies, permit globally to improve the baseline results. The accuracy criteria (ACC) is in favour of runs which limit the number of affectations. Indeed, if we use a model that assigns no category to the documents, it will ob-

tain a Macro ACC of 0,963. It is the reason why, our baseline run (run 1) and the $RCut_1$ strategy, which affect only one category, obtain the best accuracy over all our runs. In run 5, 6 and 8 several categories can be assigned to one document, and for this reason,we observe a decrease of the accuracy. The run 8 is globally worse than the others because two classes are systematically affected to each document while the average number of categories by document, estimated on the training set, is around 1.46.

On average, run 5, corresponding to the $MCut$ strategy introduced in this article, is the best of our runs. It is the best for the PRF criteria and it provides the same results as runs 7 and 8 for the MAP criteria since there is only the thresholding strategy that changes. Concerning the ROC criteria, it is slightly worse (Macro: 0.936, Micro: 0.934) than the run 7 (Macro: 0.938, Micro: 0.935).

7 Conclusion

In this article, we focused on the selection of the set of classes that will label a document for the multi-label text categorization. We propose a thresholding strategy, called $MCut$. The results obtained on the Inex XML Mining collection are encouraging. This method is compared to the commonly used approaches $RCut$ and $PCut$. $RCut_1$ and $RCut_2$ give also quite good results but they have the drawback to impose a predefined number of categories by document. Contrary to $RCut$, the number of categories for each document could be different with the $PCut$ strategy. However, this method is not suitable if we want to know the category of a new single document. So, $MCut$ seems to be a good choice because it does not make hypothesis on categories distributions and does not impose the number of category per document.

References

1. Ludovic Denoyer and Patrick Gallinari. Overview of the inex 2008 xml mining track. In *Proceedings of Initiative for the Evaluation of XML Retrieval, Lecture Notes in Computer Science*, volume 5631, pages 401–411, 2009.
2. Mathias Géry, Christine Largeron, and Christophe Moulin. Ujm at inex 2008 xml mining track. In *Proceedings of the International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX2008*, volume 5631 of *Lecture Notes in Computer Science*, pages 446–452, 2009.
3. M. F. Porter. An algorithm for suffix stripping. *Readings in information retrieval*, pages 313–316, 1997.
4. Arturo Montejo Ráez and Luis Alfonso Ureña López. Selection strategies for multi-label text categorization. In *Proceedings of International Conference on NLP, FinTAL*, volume 4139 of *Lecture Notes in Computer Science*, pages 585–592, 2006.
5. G. Salton and M.J. McGill. *Introduction to modern information retrieval*. McGraw-Hill, 1983.
6. V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.
7. Yiming Yang. A study of thresholding strategies for text categorization. In *Proceedings of the ACM SIGIR conference on Research and development in information retrieval*, pages 137–145, 2001.